published as: Waitelonis, J. & Sack, H.: Towards exploratory video search using linked data. In Multimedia Tools and Applications 53(2011), pp. 1-28, Springer Netherlands, 2011. (ISSN 1380-7501)

Multimed Tools Appl
DOI 10.1007/s11042-011-0733-1

Towards exploratory video search using linked data

Jörg Waitelonis · Harald Sack

Abstract Keyword-based search in general is particularly applicable if the searcher really knows what she is looking for and how to find it, i.e. to know the appropriate keywords to obtain the desired results. But in many cases either the objectives of the searcher are intrinsically fuzzy or she is not aware of the appropriate keywords. One way to solve this problem is to navigate and explore the search space along guided routes. In this paper we show, how Linked Open Data can be adopted to facilitate an exploratory semantic search for video data. We present a prototype implementation of exploratory video search and show how traditional keyword-based search can be augmented by the use of Linked Open Data.

Keywords Linked Open Data · Video search · Exploratory search

1 Introduction

The search for information, no matter whether you consider archives, libraries, or the World Wide Web (WWW), strictly speaking should turn out to be a win-win-situation for both the information consumer as well as the information provider. The information consumer is looking for information that the information provider supplies, while the information provider wants the consumer to find and to select his information offer. But, how can they meet?

WWW search engines as well as the subject heading catalogue of the library indicate the way for the information seekers to fulfill their information needs. While the subject heading catalogue is arranged manually—suitable keywords or keyword chains are assigned to information resources by trained experts –, sophisticated

J. Waitelonis (✉) · H. Sack
Hasso-Plattner-Institute Potsdam, Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany
e-mail: joerg.waitelonis@hpi.uni-potsdam.de

H. Sack
e-mail: harald.sack@hpi.uni-potsdam.de

Published online: 28 January 2011

algorithms are used to generate keywords automatically from (textual) information resources in the WWW. But, assigning appropriate keywords remains expert knowledge, i.e. the ordinary user hardly knows anything about the keywords, which are required to actually find a specific resource. Even worse, in the WWW the user can never be sure about the completeness and the integrity of the achieved search results.

Part of the responsibility for that situation bears the traditional keyword-based search paradigm. You have to know the appropriate keywords to find a specific resource. That's all. But, what if the prospected resource hides several hundred pages later in the list of returned results? Today, Google[1] has become synonymous for web search. The user enters a query string that might consist out of one or several keywords. Then Google's web search engine delivers (text) documents containing these keywords or multimedia documents annotated with metadata including these keywords. In the majority of cases this approach is absolutely sufficient. But, not all search engine users have the same information needs, because users might have different ways to search for information. If the search task is getting more complex, i. e. if a single document is not the answer to the user's search problem, different and more complex search strategies have to be applied. Moreover, if the user tries to achieve an overview of actually available information about a certain topic, today's web search engines are flooding search results by millions. Thus, giving the user no chance to review nothing but the first few resulting pages. Traditional keyword-based search does not consider the meaning (semantics) of the content of the underlying information and result ranking is mainly based on link popularity.

Semantic search promises to enhance keyword-based search by taking into account the actual content of the information and its semantics. By semantic annotation information resources can be related to each other, hidden and implicitly existing relationships can be made explicit. Instead of turning a small keyword spotlight towards our information universe, we can make use of all the properties of its information resources and their relationships among each other to enable the guided exploration of the search space as well as the possibility for serendipitous discovery.

Sometimes, users are looking for a specific set of documents that contains almost all the keywords of the query string (navigational searches), while in many other cases the user tries to gather information about a specific subject with no particular document in mind (research searches) [18]. In complex search tasks, the user has to retrieve some facts (i. e. documents containing those facts) first, which are required to enable further search queries solving the overall search problem. Often, the user is not familiar with the topic she is searching for, and sometimes, the user is not sure about her search goal in the first place. This kind of search often is referred to as 'exploratory search' [24].

Contrariwise to facetted search approaches aiming to further refine an original search query by clustering the search results according to common properties, exploratory search broadens the scope of the search query by suggesting associated terms, concepts, and resources. These exploratory search suggestions can be used to navigate among the entire search space and to explore the repository content by user guided browsing [45]. Thus, enabling the possibility to achieve search results the user was not looking for in the first place by serendipity.

[1] http://www.google.com/

In this paper, we address the problem of how to deploy explorative search for video data by using semantic search technology and semantic web resources. In recent years especially audiovisual media have become the predominant media of the internet. To enable content based video retrieval, high quality textual metadata have to be provided that describe the content. Most times, sufficient quality can only be achieved by time and cost intensive manual metadata annotation, while collaborative approaches deploy non-authoritative user-generated metadata, and automated video analysis is achieving progress. But, even though sufficient metadata can be provided, explorative investigations will be limited by the paradigm of keyword-based search.

The Linked Open Data (LOD) [38] project aims at making semantic data freely available to everyone and provides starting points to extract relationships among information resources. We show how to use the LOD resource DBpedia [1] to implement an exploratory search for the video search engine yovisto.com[2]. Starting with a simple keyword-based query, relationships between information instances within Yovisto's database are discovered by mapping terms with LOD resources and by utilizing their ontological structure. Thus, the user has not only access to keyword-based search results, but will also be guided by content-based associations to enable serendipitous discovery.

The major contributions of this paper are the following: We have developed several heuristics for ranking entity properties and relationships of LOD resources including efficient offline indexing for semantic video search. The most relevant properties are applied for the generation of search suggestions that are related with each other as regards content, and the quality of the overall exploratory search approach is shown by evaluation.

The paper is organized as follows. Section 2 presents related work and introduces to exploratory search, the Yovisto video search portal, and the LOD project. Section 3 details, how Linked Data can support exploratory search. In Section 4, quantitative results are discussed and in Section 5 an evaluation of how Linked Data resources complement our original video data is presented. The last section concludes the paper with a brief outlook on future work.

2 Exploratory search—foundations and related work

This section introduces semantic web technology and the prerequisites to implement exploratory search for the video search engine Yovisto by harnessing LOD. Furthermore, the concept of exploratory search and the Yovisto search engine including its ties to the Semantic Web are explained.

2.1 Semantic web and Linked Open Data

The idea of the Semantic Web was described in 2001 by Tim Berners-Lee et al. as "A new form of Web content that is meaningful to computers". It was introduced as an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation [3]. The Semantic Web

[2]Yovisto—Academic Video Search: http://www.yovisto.com/.

is about making meaningful links between heterogenous data sources (Linked Data) to enable persons and machines to explore a "web of data" [2]. The interlinking enables to navigate from one resource to other related resources from different data sources and to discover more information about them.

The Semantic Web is based on the Resource Description Framework (RDF), which is a standard model for data interchange on the Web. RDF extends the linking structure of the Web to use URIs to name the relationship between 'things' as well as to identify the things itself. Usually this is expressed in RDF triples (*sub ject*, *predicate*, *ob ject*). All RDF triples form a directed, labeled graph which edges represent a named link (predicate) between two resources, represented by the nodes [21].

The relationships and properties RDF resources may have can be specified by the vocabulary description language RDF Schema (RDFS) [12]. RDFS defines classes and properties that may be used to describe classes, properties, and other resources. Furthermore, the Web Ontology Language (OWL) facilitates greater machine interpretability of Web content than supported by RDF(S) by providing additional vocabulary along with a formal semantics [42]. An ontology is an explicit, formal specification of a shared conceptualization and defines the terms used to describe and represent an area of knowledge [17].

Compared to other structured data accessible on the Web by various APIs, Linked Data provides a single, standardized access mechanism instead of relying on diverse interfaces and result formats, which makes it highly interoperable [4]. The Linking Open Data (LOD) project aims to identify datasets in the Web that are available under open licenses, re-publish these in RDF and interlink them with each other [5]. Interlinking resources across various data sources leads to a huge network of data, referred to as the LOD cloud, currently consisting out of more than 13.1 billion RDF triples interlinked by more than 142 million RDF links (May 2010) [38].

One of the key interlinking hubs of the LOD cloud is DBpedia, the semantic counterpart of the online encyclopedia Wikipedia. DBpedia generates RDF-triples from Wikipedia infoboxes and publishes them via SPARQL [31] and RDF dump files [5]. DBpedia serves as the main source for interlinking Yovisto's metadata to put exploratory search in practice.

Linked Open Data has become one of the most popular topics among the emerging Semantic Web [5]. Correspondingly, Semantic Web resources and technologies are applied to augment the traditional search scenario. According to Guha et al. 'Semantic Search is the application of the Semantic Web to search' [18]. In this paper we present a semantically enhanced exploratory search based on LOD resources, in particular on DBpedia. A comprehensive survey on different approaches to Semantic Search is given in [23], while in [44] a formal model of ontology-based Information Retrieval in general is presented.

There are two main challenges for Semantic Search [18]: (1) the query input has to be mapped to concepts and entities [19, 25] and (2) the search domain has to be augmented with semantic content [14]. Both challenges bear the problem of solving disambiguation of homonyms. In the first challenge, this issue can be managed simply by asking the user while entering the query terms for the intended meaning, as e. g. solved by the MultimediaN E-Culture demonstrator [37] and Freebase Parallax.[3] The

[3] http://www.freebase.com/labs/parallax/

second challenge often leads to the problem of named entity recognition in textual documents, as e.g., solved by the Wikipedia Miner [27]. Alternatively, Semantic Search is often referred to as retrieval of data from the Semantic Web, as being represented by semantic search engines such as Sindice [28]. The work in this paper does not relate to this interpretation of Semantic Search.

2.2 Video search with yovisto.com

State of the art video retrieval systems use a combination of visual and textual feature extraction for search index generation and combine these techniques with machine learning procedures [10, 41]. However, the chosen feature extraction depends on domain and task characteristics and determines the quality of the retrieval system.

To enable keyword-based search in general, video search engines require content-related metadata that can be generated automatically and also manually by the user. A distinction is made between metadata created by a reliable source, such as the author or an expert (authoritative annotation) or by unreliable sources such as users and recipients (non-authoritative annotation). Metadata can be utilized on different levels of abstraction. They can describe low-level features, such as e. g., dominant color of a video frame, or a motion flow direction of a shot, and high-level features such as, e. g., the semantics of complex scenes. Furthermore, metadata can refer to the entire video resource or to spatio-temporal fragments of a resource.

Yovisto is a video search engine specialized in academic lecture recordings and conference talks. Unlike other video search engines, Yovisto provides a time based video index, which allows to search within the videos' content. Yovisto's index is built up from fine-granular time-dependent metadata. Automated analysis techniques such as scene detection and intelligent character recognition are used for metadata generation [35]. In addition, time dependent collaborative annotation enables the user to annotate tags and comments at any point within a video [34]. Yovisto allows faceted search to filter and to aggregate the search results, which simply enables a refinement or further filtering of the already achieved the search results, but not an expansion of the query in the sense that the search scope should be broadened in an appropriate way. Broadening the scope and suggesting nearby related search alternatives is the task of an exploratory search feature.

Yovisto's metadata is encoded in the standardized and interchangeable metadata description framework MPEG-7 to ensure interoperability [13]. Currently, Yovisto provides more than 10.000 videos (ca. 9.500 hrs.) with 2.1 million index keywords and 23.000 user generated annotations.

To facilitate a suitable application programming interface (API) for mashup web applications, Yovisto's metadata is published in RDF format, being embedded as RDFa in the webpages and also accessible via a RDF triple-store.[4] Following the Linked Data principles Yovisto data is mapped to the LOD cloud [45]. To achieve this, an OWL-DL ontology has been defined to represent the Yovisto data structure[5] reusing already existing ontologies and vocabularies to enable interoperability (DublinCore [20], FOAF [9], tag-ontology [33], MPEG-7 Ontology for the MPEG-7 XMLSchemas [16]).

[4] http://sparql.yovisto.com/

[5] Yovisto ontology: http://www.yovisto.com/ontology/0.9/.

We extend the search capabilities of Yovisto by adding an exploratory search feature that enables the user to browse the content of the underlying video repository in a multi-faceted way. In difference to popular recommender systems [15], our novel approach is neither based on logfile analysis and statistical usage analysis of content popularity [6], nor on similarity-based methods such as query by example [22].

2.3 Exploratory search

In contrast to traditional keyword-based search, exploratory search assists the user in exploring the data space to improve search experience. Thereby, the user is able to navigate the search space, as well as to reorganize the content and the user interfaces for her own needs with appropriate interactive elements. While searching, the user can choose between alternatives, move along paths, and move back to choose an alternative way. To implement explorative search, the underlying data needs to be fully made accessible. Relationships between associated resources have to be made explicit to let the user navigate along them. Typically, there are different kinds of relationships, e. g. resources belonging to the same category, authored by the same person, etc. One way to establish a simple exploratory search is to reorganize and to filter the search results according to these relationships by so-called faceted search [30].

For example, Schreafel et al. developed mSpace, a multi-column faceted spatial browser for multimedia data [36]. Petratos described facets as conceptual categories, which are created to organize the presentation of all available data into an easy to view concise set of conceptual groups [29]. Furthermore, explorative search also means to discover new associations and new kinds of knowledge.

Marchionini differentiates between lookup, learn and investigation search [24]. Driven by straight fact retrieval and an analytic search strategy, lookup search is the most basic type of search. Moreover, learning search involves multiple iterations and requires cognitive processing and interpretation of the returned sets of objects. Requiring strong human participation in a continuous and exploratory process, Marchionini considers learn and investigation search to be exploratory search. Active user involvement in the search process and uncovering new connections between resources is an essential characteristic of exploratory search. This implies new search interfaces with exploratory navigational components to be able to delve deeper into a repository than before.

Exploratory search can be applied to any type of resource. Especially time-dependent multimedia such as video facilitates the visualization of different views on the media. The problem with time based media (e.g. video or audio) in exploratory search is that in most cases there is no textual representation of it's content available. Content information has to be extracted from media data by automated feature analysis such as Optical Character Recognition (OCR) or Automated Speech Recognition (ASR). Feature analysis often leads to insufficient results, causing further problems with the analysis of the achieved textual representation, as e.g., valid entity mapping for imperfect text is almost impossible. Mapping text to semantic entities also requires contextual analysis. For video the context can be defined by a single coherent scene or shot. Furthermore, user interfaces to browse, search, and view time based media differs from user interfaces for textual documents, i.e. the visual or auditive information in video and sound is used to create navigational components

and content browsing tools. For example, Christel discusses video storyboards as exploratory interfaces and how to move beyond fact-finding by investigating multiple views and visual exposition of metadata and multimedia surrogates [11]. Basically, storyboards give an overview of the visual characteristics of a video. But, for visually homogeneous video data, as e. g., in the recording of a conference talk or a lecture, it is difficult to deduce the content from its visual features only.

Other systems for exploratory search and facet browsing user interfaces are 'SIMILE seek'[6] for browsing email folders, the general purpose facet browser of 'flamenco project' [47], or the 'elastic lists' demonstrator [43], that uses the same dataset. Those user interfaces have in common to work on selected datasets, not derived from RDF sources. They mainly focus on how to enable browsing on the user interface level. Our work is more focused on showing that unreliable, heterogenous data sources, provided by LOD can be utilized to enable exploratory search at all, and that the structure of LOD provides useful characteristics, which can support an exploratory search or facetted browsing feature. Furthermore, the approach tries to create the suggestions in a fully automated way, compared to 'freebase'[7] for example, where the displayed information is assembled manually through collaborative creation of user-created 'views' for a specific topic or types. This works well, if there is a huge community working on it and if the topics are general domain. Since the freebase views are created manually, they could serve as a reference for an evaluation.

This section has introduced exploratory search, its foundations and presented the state-of-the-art. Furthermore, it was referred to Semantic Web technologies, and the video search engine Yovisto.

3 Using linked data to enable exploratory search

This section deals with the process of exploratory semantic search and its implementation in the video search engine Yovisto. To enable exploratory search, heuristics have been developed to rank existing relations (properties) between DBpedia entities to determine their importance. To begin with, an introductory example is presented and the functionality of the prototype graphical user interface (GUI) is explained.

3.1 The user interface for exploratory search

The graphical user interface (cf. Fig. 1) is designed to comprise three main areas: the *direct search results* in the center column including optional geographical information displayed in a map on top of the search results, the *facet filter* on the right, and the *exploratory search navigation* on the left. The search results include a timeline, which shows the automatically generated temporal segmentation of the video results including highlighted segments indicating search hits. The facet filter allows to narrow the search results according to the type of resource, the scientific category,

[6] http://simile.mit.edu/seek/

[7] http://www.freebase.com/

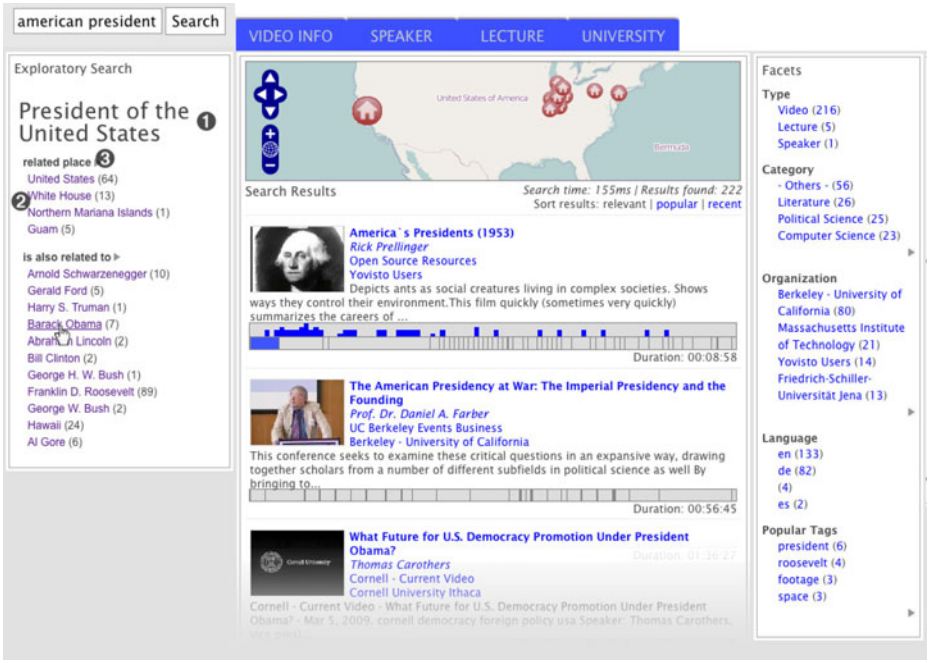


Fig. 1 The exploratory search GUI showing related entities for 'american president'

the issuing organization of the video, and the language of the video, as well as popular user tags attached to the video segments.

Exploratory search aims to broaden the scope of search by suggesting related terms, concepts and resources. Our approach uses LOD resources to support the search process by exposing additional information about indexed resources in Yovisto, which are semantically interrelated to the users search query.

Figure 1 depicts the result of a query for *'american president'* that is mapped to the DBpedia entity 'President of the United States'. The exploratory search GUI suggests a list of related entities. When the user enters a query string, the labels of the mapped entities (1) are shown distinctly below the search input field followed by all related entities (2) grouped by their connecting properties (3). Next to the related entity labels a number in brackets denotes how many video resources for this particular entity exist within the Yovisto video repository.

By clicking on, e. g. '*Barack Obama*' in the exploratory search GUI, a new search is issued and the GUI switches to the newly selected entity showing its related entities and properties (cf. Fig. 2). This supplementary information includes, as e. g., related places (birth place, work place, etc), predecessor and successor in the presidential office, or Barack Obama's residence.

To retain previous actions, a history list (4) provides links to previous searches. Optionally, the user may activate an additional preview of the search results evoked by a related entity when clicking on it (5). Moving the mouse pointer over these previews causes a popup to show brief information about the video resource (6).

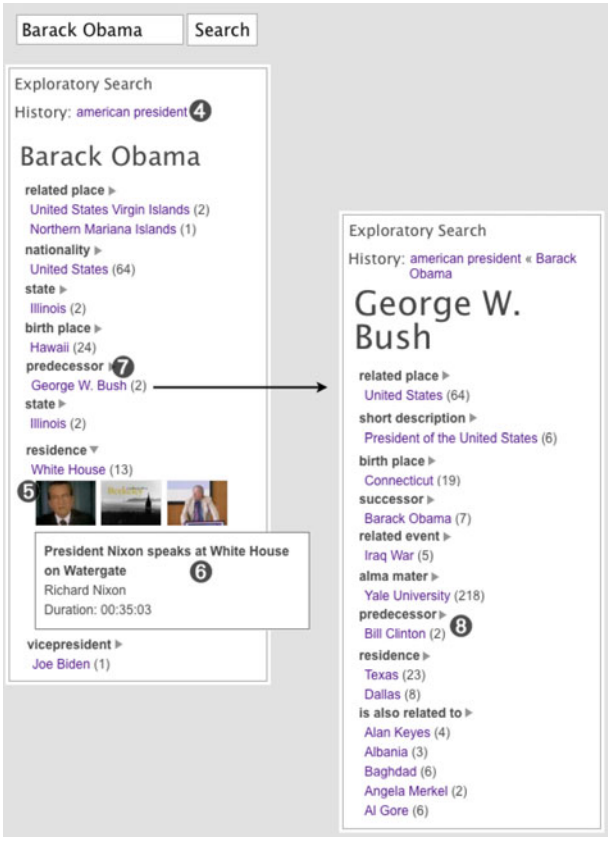


Fig. 2 The exploratory search GUI showing related entities for 'Barack Obama' and 'George W. Bush'

In the example some DBpedia properties such as *'predecessor'* (7, 8) have the characteristic trait to connect entities of the same type. They allow to move 'hand over hand' from one entity of a distinct category to the next, which enables the user to quickly exploit the information of individual entities.

Almost any DBpedia information can be made available as supplementary information to propose ancillary search possibilities to the user. But, DBpedia provides way too much information about each single entity to be displayed to the user in total. Hence, we have to determine the most important or relevant information. Therefore, we have developed heuristics based on statistical and structural characteristics of the DBpedia RDF graph. This heuristics enable to rank existing relations (properties) between DBpedia entities to determine their importance. Before explaining the applied heuristics in detail, the overall processing workflow is explained and formalized in the following section.

3.2 Process workflow of exploratory search

To enable exploratory search as outlined in the previous sections a three-step procedure has been devised: (1) each keyword of the user query is mapped to one or more DBpedia entities with the help of a gazetteer dictionary, (2) mapped entities

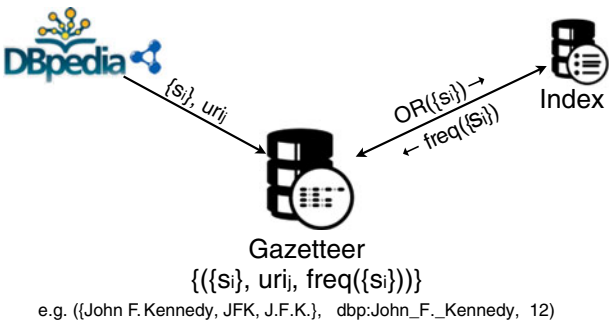


Fig. 3 Using synsets for the generation of the gazetteer

are cross-checked with the Yovisto repository, and (3) for each resulting entity the most important related resources are determined.

The following sections encompass the workflow stages in detail.

3.3 Mapping queries to entities

To map queries to entities a *gazetteer* dictionary is generated form the DBpedia data sources and stored in a database (cf. Fig. 3). The processing of this intermediate data is rather computational intensive. Therefore, computation is performed offline to maintain fast query responses. The gazetteer is implemented as a named-entity list and comprises a list of synonyms (synset) $\{s_i\}$ for every DBpedia entity being represented by its URI. Furthermore, the number of hits, when searching for the synset in the Yovisto search index, is stored as $freq(\{s_i\})$. Hence, the gazetteer is defined as set of (synset, URI, hits)-tuple: $\{(\{s_i\}, uri_j, freq(\{s_i\}))\}$.

3.3.1 Creating synsets

To create the synset for an entity, different sources in DBpedia have to be utilized. The most reliable DBpedia source for entity mapping is the DBpedia URI-suffix. The URI-suffix denotes the string, which remains after removing the prefix 'http://dbpedia.org/resource/' from the URI and replacing underscores '_' by single whitespaces. In the majority of cases, the URI-suffix denotes the entity most suitably. In addition, literals of the DBpedia property *rdf:label* are eligible, if they differ from the URI-suffix. Table 1 shows an example for synonyms determined for the given entity 'John F. Kennedy'. In many cases *rdf:label* provides labels

Table 1 Synonyms generated for the entity: http://dbpedia.org/resource/John_F._Kennedy

Synonym	Type
John F. Kennedy	URI-suffix
John F. Kennedy	label
John Fitzgerald Kennedy	label
John Kennedy	redirect
J. F. K.	redirect
JFK	redirect
35th President of the United States	redirect
John f kenedy	redirect

(nametags) in different languages, but in some cases, there is no *rdf:label* provided at all. Furthermore, so-called 'DBpedia redirects' are an additional source for synonyms. A redirect occurs, if a widely accepted different spelling or a common misspelling for the resource exists. Redirects are identified by the DBpedia property 'http://dbpedia.org/property/redirect'. Finally, the URI-suffix or labels of a redirect object are taken into account for synonyms. We have applied the DBpedia dump files to generate synsets for more than 3.1 million different URIs.

3.3.2 Index alignment

What still remains is to determine $freq(\{s_i\})$ by computing how often the synset is represented within the Yovisto search index. Hence, we need to map the computed synsets to the Yovisto index data. To obtain this, an index search comprising the related entity synset is issued (cf. Fig. 3, right). For this query all synonyms interlinked with boolean OR form a new query string. This ensures the completeness of the result list, consequentially increasing recall, but in some cases at the expense of precision. The index request returns the number of results for a direct search based on the entity synset, which is additionally stored in the gazetteer list.

We have now created an named-entity list to map queries and Yovisto metadata to DBpedia entities. To enable exploratory search in Yovisto, implicitely given associations between videos based on their content have to be made explicit. Therefore, we need to find relations between the entities the videos comprise.

3.4 Discovering related entities

Related entities are determined by the application of a heuristic based ranking for all the associated entities of a given entity e. The predefined heuristics are applied to all 3.1 million entities in DBpedia and enable to filter out low ranked, i. e. less important associations. For a given entity e a related resource is initially defined as:

$$r = (uri, \{s_i\}, l, p, pl, rank)$$

with *uri* representing the URI of the related entity r, $\{s_i\}$ the synset for the URI, l a human readable entity label, p the property the given entity is connected with the related entity, pl a property label for display, and *rank* the ranking among all related resources of the given entity e. The rank of an related entity depends on which and how many heuristics have ascertained the related entity as to be of importance.

Figure 4 schematically shows the process of generating related entities. For all entities of DBpedia related entities are determined by heuristics. This results in a list of related entities for every DBpedia entity e. For exploratory search in Yovisto the related entities only comprise entities that are also present in the Yovisto search index. Therefore, the entity frequency provided by the gazetteer is also included in the computation. Hence, the related entities list is defined as:

$$\{(uri_i : \{r_1, ..., r_m\}\} \text{ with}$$

$$r_j = (uri_j, \{s_i\}, l, p, pl, rank, freq(\{s_i\})) \text{ with } freq(\{s_i\}) > 0.$$

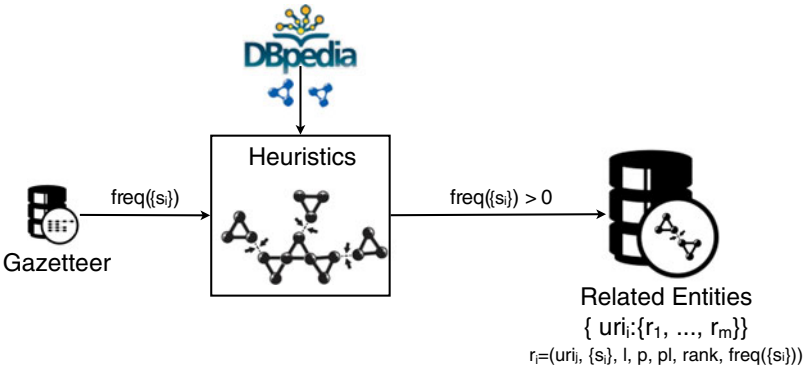


Fig. 4 Identification of related entites

Example: For the entity e, representing the Person 'John F. Kennedy', let $uri_i = 'dbp:John\_F.\_Kennedy'$ be the URI of e. Among the list of related entities r_j of e there might be:

$$\begin{aligned} r_j = (uri_j &= 'dbp:Lyndon\_B.\_Johnson', \\ \{s_i\} &= \{'Lyndon\ B.\ Johnson', 'Lyndon\ Johnson', 'LBJ'\}, \\ l &= 'Lyndon\ B.\ Johnson', \\ p &= dbp:successor, \\ pl &= 'Sucessor', \\ rank &= 0.456, \\ freq(\{s_i\}) &= 8). \end{aligned}$$

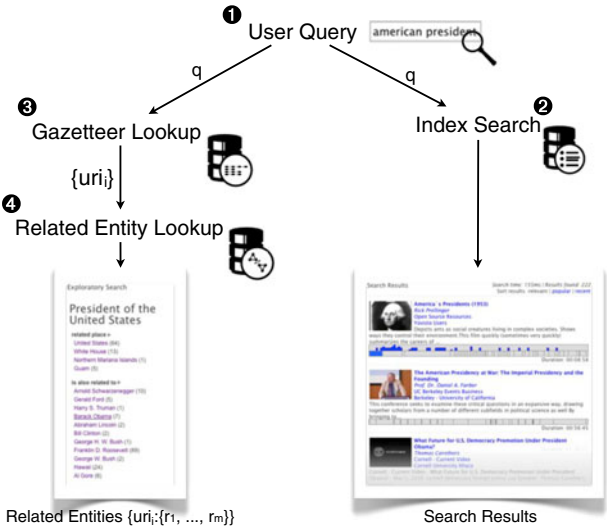


Fig. 5 Overall process workflow with related entities recommendations

Table 2 Property ranking heuristics and their rankings

No.	Heuristic
1	Frequency-based (F)
2	Same-RDF-type (R)
3	Events (E)
4	Places (P)
5	Dual properties (D)
6	Backlinks (B)
7	Wikilinks (W)
8	Inlink (I)
9	Lists (L)
10	Categories (C)
11	Ontology (O)

Finally, Fig. 5 outlines the entire exploratory search process. The user query (1) is propagated to the search index (2) and to the entity mapping (3). The search index creates the regular search results. The gazetteer lookup (3) returns a set of URIs mapping to the query string. For every URI the related entities are looked up (4) and are displayed to the user.

We now discuss the heuristics one by one, which are used to identify important resources and properties.

3.4.1 Heuristics for property-ranking

The heuristics are used to rank the properties of DBpedia entities according to their importance. An entity ranking value can be derived directly by aggregating the rank of its properties. Hence, ranking the properties only is sufficient. An overview of the developed heuristics is presented in Table 2.

1. Frequency-based heuristic (F) This heuristic is based on the assumption that the more often a property occurs on instances of a specific category or type, the more relevant it is for this category in general. As input for this heuristic the frequency of RDF properties used in conjunction with concepts of a specific RDF type (rdf:type) or SKOS[8] category (skos:subject) in DBpedia are taken into account. Table 3 shows the frequencies of various properties for all entities with skos:subject *Category:Presidents_of_the_United_States*. For exploratory search we suggest only related entities connected to the high frequent, i. e. most important (popular) properties. If a resource belongs to more than one skos:subject, the frequencies for the properties of each category are added up.

2. Properties based on same rdf:type (R) Starting off with the idea to consider resources of the same category being relevant to each other, properties connecting resources of the same `rdf:type` are considered to be important, because they are semantically closely related. The same holds for the resources being connected by these properties. To determine important properties, all connected resources (objects) of the same category have to be verified against interlinked instances.

[8]Simple Knowledge Organization Systems, a family of formal languages designed for representation of structured controlled vocabulary [26].

Table 3 Properties and occurrence frequencies of DBpedia entities with skos:subject *Category:Presidents_of_the_United_States*

No.	Property	Frequency
1	battles	29
2	predecessor	10
3	successor	10
4	navy	10
5	order	10
6	alongside	9
7	state	9
	...	
22	list	6
23	name	6
24	office	6
25	oldstyledatedyProperty	6
26	years	6

Figure 6 illustrates the following example: Albert Einstein and Alfred Kleiner are both scientists. Albert Einstein is a scientist as well as an American vegetarian. According to DBpedia, Bill Cosby is an American vegetarian, too. The property `dbpedia:doctoralAdvisor` is identified as relevant, because it connects both instances of the category `dbpedia:Scientists`. In contrast, the other American vegetarian (Bill Cosby) is not tightly coupled to Albert Einstein, because there are no properties connecting both of them directly.

The following SPARQL query identifies properties and corresponding objects of the same category.

```
SELECT DISTINCT ?p, ?o WHERE {
    <uri> ?p ?o.
    <uri> rdf:type ?type.
    ?o rdf:type ?type.}
```

3. Properties to Events (E) Selected DBpedia categories are considered to be of general importance for our application. Among these are, as e. g., `dbpedia:Place`, and `dbpedia:Event`. Yovisto is specialized on academic content, which always comprises information about places as well as events. Furthermore, most things of interest for the arbitrary searcher are related to locations, or events. Properties directing to instances of these special classes are pursued and presented to the user.

4. Properties to Places (P) Similar to the event based heuristic E, this heuristic considers properties referring to places.

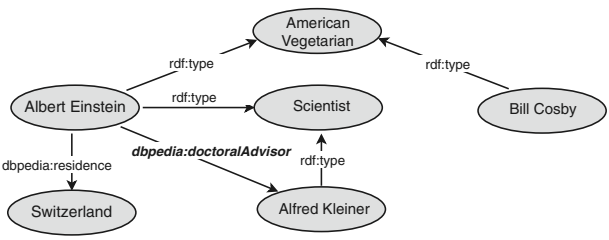


Fig. 6 Property between classes of same rdf:type

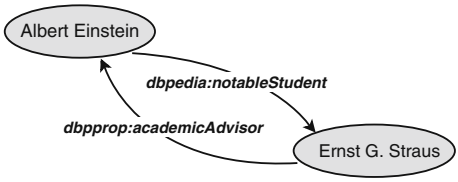


Fig. 7 Dual properties

5. Dual properties (D) Resources that are both connected explicitly with each other via reversal relations are considered to be important, because there is evidence that both resources have similar characteristics. For example, Fig. 7 depicts Albert Einstein and Ernst G. Straus. Each one is connected to the other with a different property. Both properties `dbpedia:academicAdvisor` and `dbpedia:notableStudent` are connecting the resources in both directions and therefore we deduce evidence for a closer relationship. The properties are not defined as *inverse* properties. But each time the one property exists, the other property exists, too.

The following SPARQL query selects properties and resources where this duality applies to:

```
SELECT DISTINCT ?p1, ?p2, ?o WHERE {
        <uri> ?p1 ?o.
        ?o ?p2 <uri>.
        FILTER(?p1 != ?p2).}
```

6. Backlinks (B) The property `dbpedia:wikilink`[9] represents an untyped HTML-hyperlink between two Wikipedia articles. If Wikipedia article `<A>` contains a link to article `<B>`, there will be an RDF-triple **`<A>`** `dbpedia:wikilink` **`<B>`**. Objects, which have a bidirectional wikilink are considered to be of higher relevance and closer related to the subject. It is assumed that resources connected with bidirectional wikilinks are highly interrelated (c f. Fig. 8).

The following SPARQL query selects objects for a given subject `<uri>`, which have a bidirectional wikilink.

```
SELECT DISTINCT ?o WHERE {
        <uri> dbpedia:wikilink ?o.
        ?o  dbpedia:wikilink <uri>.}
```

7. Unidirectional wikilinks (W) Similar to bidirectional wikilinks, unidirectional wikilinks are indicating a semantic interrelation. But this relationship is considered to be weaker than bidirectional wikilinks.

8. Incoming-Wikilinks (I) This heuristic specializes the Wikilink (W) heuristic, but considers only those Wikilinks, which are incoming to the focus resource.

9. Lists (L) Some properties link to resources, representing aggregations of other resources such as lists. These resources can be identified, if their URI suffixes start

[9]Currently only supported by DBpedia dump files, and not by the SPARQL endpoint.

Fig. 8 Bidirectional wikilinks

with the string `'List_of_'`, such as in `dbpedia:List_of_Nobel_laureates`. In most cases, Wikilinks refer to these kind of resources. Hence, this heuristic is a specialization of the wikilink (W) heuristic.

10. Categories (C) This heuristic stands for the SKOS *subject* property. Usually, this property refers to resources that represent Wikipedia categories. Within the dump files of the German language version of DBpedia, categories are connected via Wikilink properties and not via SKOS subject.[10] Categories enable the user to cognitively classify and structure the information as well as to find other instances from the same category during the exploratory search process. Usually, entities belonging to the same category are semantically related. Therefore, we consider categories as appropriate for recommendation.

11. Ontology (O) This heuristic takes into account the RDF-type property. It refers to classes from the DBpedia ontology. Compared to the R heuristic, which operates on instance level, this heuristic is working on the ontology level. The recommendation of associated classes enables the user to resolve ambiguities and also to find other instances of the same class.

The proposed heuristics are based on the assumption that interconnected entities in DBpedia are more closely related to each other than entities without any RDF graph connection.

The selection of recommendations for exploratory search depends on a ranking of the heuristics. The individual heuristics are ranked in the same order as they appear in this section. Due to restrictions of the GUI only a limited number of recommendations can be displayed to the user. For reasons of efficiency, the most relevant heuristic (frequency-based heuristic) is computed first. If the result is too sparse to be displayed in the GUI, the next heuristic will be computed, and so forth. Computing the individual heuristics only on demand according to their ranking is necessary, because it is too time consuming always to compute the results of all heuristics. The chosen ranking has been confirmed in the evaluation of the performance of the heuristics in Section 5.

The next section presents statistics about the preprocessing workflow before focussing on the evaluation in Section 5.

4 Preprocessing workflow—insights and statistical results

In this section statistics about the preprocessing workflow are given and performance issues are pointed out.

[10] Refers to version 3.5.1 of DBpedia.

Table 4 Distribution of related resources depending on heuristics

Heuristic	# of occurence
Wikilink (W)	14.340.566
Backlink (B)	3.247.761
Frequency-based (F)	1.341.769
Places (P)	531.801
Same-RDF-type (R)	309.093
Events (E)	25.933
Dual properties (D)	24.783

For overall 3.1 m DBpedia entities we determined more than 6.5 m terms that are stored as synsets in the gazetteer list. For all 3.1 m DBpedia entities 19.9 m mappings to related other DBpedia entities were found with the help of the proposed heuristics. Table 4 shows the distribution of related resources found according to the different heuristics[11]. Most associations could be found with the Wikilink (W) heuristic, while Dual properties (D) do only show up rather rarely, which corresponds to previous results [46].

Figure 9 exemplifies the distribution of the number of related resources for the heuristics Wikilinks (W), Backlinks (B), Events (E), and Places (P). Less than 10 related resources are assigned to each of the more than 80 k terms by the Wikilinks heuristic. Places and Events lead in the majority of cases to less than three related resources per entity. It is notable that the distribution reflects a power law, which corresponds to theoretical expectations.

More than 165 k DBpedia entities are directly mapped with the Yovisto search index and more than 582 k related entities have been found for them, where as 108 k of them are mapped back into the Yovisto search index. In total, more than 1.1 m new links have been generated among Yovisto videos at all. Figure 10 illustrates the DBpedia entities and their links to related entities including frequencies.

4.1 Performance issues

For the proposed explorative search, processing online queries with DBpedia is very time consuming. Especially, if many associated resources have to be retrieved the processing of a single term might take up to one minute or more. Hence, an offline processing has been set up to process every term beforehand. Furthermore, due to the numerous SPARQL queries, a local copy of DBpedia has been set up. This was not only necessary because of performance issues, but also because the dump files of DBpedia do contain more useful information than the available online repositories (e. g., Wikilinks in DBpedia). Furthermore, queries to the online version of DBpedia are limited to results of at most 1.000 RDF triples.

Regardless of their nature, be it keyword-based, be it multimedia, or be it exploratory, to measure the effectiveness of a search engine implementation a reasonable evaluation has to be performed. The following section shows a qualitative evaluation by user centric assessment of the GUI and another quantitative evaluation of the proposed heuristics.

[11] Currently, not all heuristics have been processed completely for the production system.

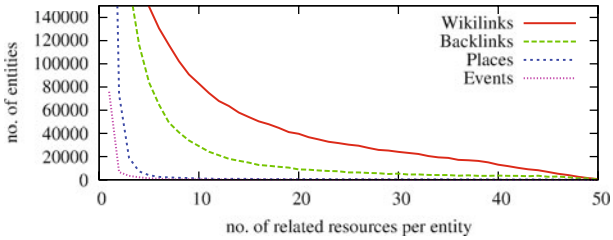


Fig. 9 Related resources found by Wikilinks (W), Backlinks (B), Events (E), and Places (P) heuristics

5 Evaluation

In this section an evaluation method for exploratory search scenarios and the proposed heuristics is presented including a discussion of the results. Evaluation of traditional information retrieval systems is based on rather quantitative than qualitative measurements of the achieved retrieval results. The retrieval results are compared to a ground truth resulting in an objective assessment of the achieved quality, i. e. by statistical classifications such as recall and precision.

While the evaluation of traditional information retrieval systems do focus on quantitative measures for the quality of retrieval results, the evaluation of exploratory search strongly depends on qualitative measurements.

Concerning the definition of exploratory search, the user does not always exactly know what documents she is looking for. This originates from the fact that the user may not be familiar with the search topic. Perhaps she does not know, where to begin and where to end the search, as well as she might not be sure about the search goal in the first place. Thus, it is rather difficult to define an objective ground truth for given exploratory search tasks, because individual search strategies, motivations, and interests cause ground truth also to depend on the eye of the beholder. In this case, quantitative evaluation measures such as precision and recall are less significant for exploratory search tasks than qualitative measurements, such as user satisfaction with the achieved search results and user experience during the search process.

The focus of evaluation strategies from the well known TrecVid benchmarks lies on pure system evaluation. Evaluation based on direct user involvement, referred to as 'User evaluation' is explicitly mentioned as out of scope for these benchmarks [40].

To demonstrate the added value of newly implemented retrieval features execution of the same evaluation task is suitable with and without application of the specific retrieval feature. The differences between the resulting measurements point out the effect of the new retrieval feature. Singh et al. apply this evaluation strategy in [39]. We have adopted this approach for our evaluation to demonstrate the usefulness of exploratory search in Yovisto. The motivation to use this strategy lies in the subjective and investigative nature of exploratory search [24], which

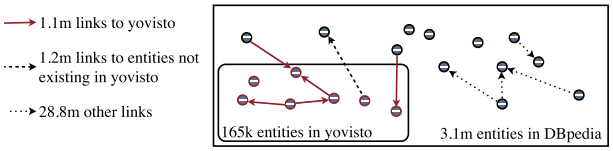


Fig. 10 DBpedia entities and the links to related entities with 19.9 links in total

makes it difficult to determine an objective ground truth for reference. Therefore, we apply additional qualitative evaluation measures by monitoring user satisfaction throughout the work task, as proposed in [32].

In [8] a framework for evaluation of interactive information retrieval systems is presented, where the user task is formulated in a cover story leading to the work task and finally to the actual search task. Two evaluation strategies are compared, which distinguish multiple types of relevance, inter alia, so-called 'Situational Relevance', which reflects the dynamic nature of relevance [7]. Situational relevance also applies to exploratory search scenarios, where the user's relevance scale may be influenced by the receipt of new information.

To proof the suitability and effectiveness of our proposed heuristics as a basis for exploratory search, we have manually created a ground truth to compare the results of the heuristics with. Thus, we have determined precision and recall in a quantitative evaluation with the outcome to find the best combination and a ranking of the heuristics. To show the usefulness of the exploratory search feature, we have additionally conducted a user centric evaluation to measure user satisfaction. Both evaluations are discussed in detail in the following sections.

5.1 Quantitative evaluation of heuristics

Before evaluating the effectiveness of exploratory search in Yovisto, we have conducted a quantitative evaluation of our proposed heuristics to measure the impact of recommendations of the heuristics independently from the usability of the GUI. The heuristics are applied to determine the most relevant resources associated with a given DBpedia entity. A ground truth to compare with should comprise the most relevant resources for this entity. But, relevance is a highly subjective sentiment of the user and also depends on context and pragmatics. Hence, relevance cannot be decided by a single user.

Therefore, to create an objective ground truth a multitude of different users has to contribute to collaboratively generate a dataset of sufficient size. To achieve this task we have selected a sample of 129 distinct entities from DBpedia and asked 72 users to specify the most important facts and associations about these entities. To reach as many test candidates as possible, we have set up a simple web application to let the users accomplish the objective online. Because not all test candidates are familiar with all entities of our sample, we also displayed the corresponding Wikipedia articles to help the user in finding the most important facts. Next to the displayed Wikipedia article we provided 10 empty text boxes to fill in the requested facts in the order of relevance. To enable a straight mapping to DBpedia resources, we have equipped the text fields with an auto-suggestion feature. The proposed suggestions are compiled from the entity's available RDF description. In particular, we suggested all resources directly connected to the current DBpedia entity by displaying their resource labels generated from URI suffixes. This auto-suggestion feature was necessary to avoid a subsequent error-prone disambiguation. Figure 11 shows the GUI of the evaluation web application including text fields and suggestions on the right side.

The sample of 129 DBpedia entities is presented to every user in random order. Not all user have processed all 129 items. Finally, we were able to generate a ground truth comprising 115 distinct DBpedia entities. In total, 5.225 entity-resource assignments were made, which results in 2.372 distinct user selections after replacing

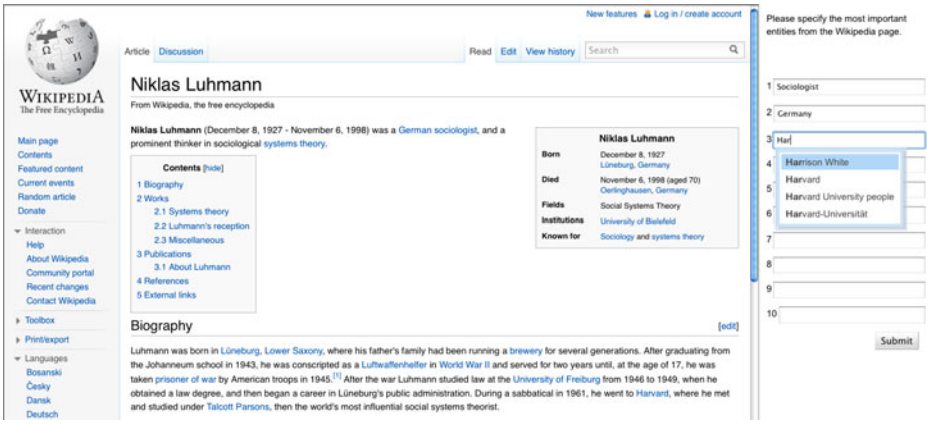


Fig. 11 The evaluation user interface for the entity 'Nicolas Luhman'; the auto-suggested labels are representing potentially related DBpedia entities

DBpedia redirects with their designated resources. This shows that a great number of assignments were made by more than a single user.

The following observations have been made and will be discussed in detail:

- How often does a heuristic generated triple occur in the ground truth (recall)?
- How well does a heuristic cover the ground truth selection of the users (precision)?
- How can we optimize the interplay of heuristics to achieve optimal results?

To answer the first two questions, we have investigated the intersection of the generated data and the ground truth (c.f. Table 5). The heuristics that achieved the best results for precision are the *Same-RDF-type (R)* and *Frequency-based (F)* heuristics. The heuristics that achieved the best results for recall are *Wikilinks (W)*, *Inlinks (I)*, and *Backlinks (B)*. The recall 0.804 of Wikilinks heuristic was expected, because for nearly every property an entity is involved with, a corresponding Wikilink exists. The conclusion of the first observation is that there is not a single 'perfect' heuristic and that high precision can only be achieved at the expense of recall and vice versa. High precision and low recall means that the heuristic provides only a few suggestions, but the suggestions are fairly relevant. However, to increase recall while preserving the achieved precision, the heuristics have to be combined with each other. For this purpose we computed the intersection of every subset of the power set of all heuristics.

None of these subsets performed better in both precision and recall simultaneously than the original heuristics separately. But some intersections were able to achieved higher precision such as the 'Frequency-based' (F) heuristic and the 'Same-RDF-type' (R) heuristic. In combination with the results achieved by the individual heuristics we were able to defined a ranking of the heuristics according to the achieved precision values (c.f. Table 6).

Based on this ranking, we have determined the top 20 suggestions for every DBpedia entity (with the intent to finally present them to user in the exploratory search GUI). To achieve this, from every heuristic/intersection a maximum number

Table 5 Comparing individual heuristics with ground truth

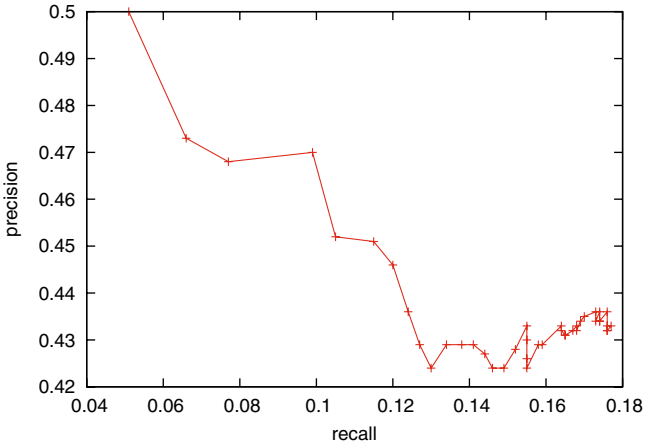
	Recall	Precision	Heuristic selected	Intersection
Backlink (B)	**0.358**	0.105	8,120	849
Categories (C)	0.011	0.017	1,565	27
Events (E)	0.007	0.006	2,883	17
Inlink (I)	**0.414**	0.003	356,574	981
Same-RDF-type (R)	0.037	**0.349**	252	88
Dual properties (D)	0.020	0.126	381	48
List (L)	0.002	0.010	399	4
Ontology (O)	0.024	0.085	662	56
Places (P)	0.128	0.004	67,638	304
Frequency-based (F)	0.079	**0.342**	547	187
Wikilink (W)	**0.804**	0.044	43,619	1,908

i of entity items were chosen. The resulting set is filled up with entity items from the remaining heuristics until 20 suggestions are reached in total per entity. Then, the result set of 20 suggestions has been intersected with the 20 most frequently selected entity items from the ground truth. Precision and recall have been computed depending on the parameter i (c.f. Fig. 12). The most promising results have been achieved in the range of $10 < i < 15$.

The evaluation shows that every heuristic performs differently. Wikilinks (W) and Backlinks (B) appear to provide the highest recall but only low precision. On the contrary, Frequency-based (F) and Same-RDF-type (R) based heuristics can be considered to be more relevant, because of better results in precision. Depending on precision and recall a ranking can be determined to organize the execution of the heuristics in an optimal order. This ranking can be refined by inclusion of the

Table 6 Best performing intersections of heuristics ordered by precision

	Recall	Precision	Heuristic selected	Intersection
F ∩ P ∩ D	0.0025	**0.5455**	11	6
F ∩ P	0.0156	**0.5286**	70	37
F ∩ E	0.0004	**0.5000**	2	1
F ∩ P ∩ I	0.0084	0.4651	43	20
R ∩ F ∩ W ∩ P	0.0051	0.4444	27	12
R ∩ W ∩ P	0.0164	0.3861	101	39
R ∩ W	0.0337	0.3571	224	80
F ∩ W ∩ I	**0.0388**	0.3433	268	92
F ∩ I	**0.0409**	0.3415	284	97
F ∩ B	0.0371	0.3398	259	88
R ∩ W ∩ I	0.0245	0.3372	172	58
R ∩ W ∩ P ∩ I	0.0126	0.3371	89	30
W ∩ P ∩ D	0.0046	0.3333	33	11
R ∩ F	0.0160	0.3304	115	38
R ∩ I	0.0257	0.3297	185	61
R ∩ P ∩ I	0.0135	0.3265	98	32
R ∩ F ∩ D	0.0067	0.3265	49	16
R ∩ F ∩ W ∩ D	0.0063	0.3261	46	15
F ∩ W	**0.0582**	0.3136	440	138



i	recall	precision
1	0.051	0.500
3	0.077	0.468
4	0.099	0.470
5	0.105	0.452
8	0.124	0.436
10	0.130	0.424
12	0.138	0.429
15	0.146	0.424
17	0.152	0.428
20	0.155	0.426

Fig. 12 Precision and recall results of combined heuristics depending on the number of items i selected from each heuristic/intersection

previously computed intersections of heuristics. To suggest 20 recommendations from each heuristic 10 to 15 items should be selected to preserve the achieved results. Note that higher ranked heuristics only provide suggestions very rarely.

We have compared the results of the heuristics to a manually created ground truth to provide a quantitative evaluation. Now, we carry on with complementing our evaluation by measuring the satisfaction of the user and the usefulness of the new exploratory search feature.

5.2 Qualitative user centric evaluation

We have made up 9 different search scenario tasks to be solved by test users. For exploratory search, the tasks have to be formulated in a way that there is most likely no direct answer possible. Moreover, the tasks must involve an iterative search strategy, where the answers being achieved in the first step are applied as input to the second search step, etc. E. g., instead of asking 'find videos about Barack Obama' we asked the user to retrieve videos about all US presidents. Thereby, in the first place, the user has to find out the names of the former US presidents before retrieving videos about them. To compare exploratory video search with traditional video search, we presented the same search tasks to different users, where we asked one group to solve the tasks with the help of the exploratory search feature, while the other group (control group) had to solve the task without the exploratory search feature, i. e. without the exploratory search sidebar activated in the GUI.

Of course we first had to figure out, which retrieval topics were really suited for the scope of the Yovisto video repository. The resulting evaluation tasks are listed in Table 7. For the evaluation we did not limit the time required for each single task, but left it to the user to decide when to finish. Not all tasks were processed by every test person. While working on the retrieval tasks the test persons were asked after every partial search step, if they think it is still possible to achieve the search objective in this search session, to gather information about the motivation of the test person. The

Table 7 Search tasks for evaluation

1.	Which other scientists did Albert Einstein know personally in the 1920s and on which event he might got to know them?
2.	Which philosophers build on the theories of the greek philosopher Plato?
3.	Find videos with information about the German chancellors from 1949 until today.
4.	Find videos about celestial bodies of the solar system.
5.	Find videos about film directors.
6.	Which videos contain information about US federal states?
7.	Find videos about the founders and main promotors of the Enlightenment movement.
8.	Find videos about cities of the Hanseatic League.

evaluation interface also provided the possibility to select and mark relevant videos among the retrieval results according to the test person's opinion. The decision, if a video in the retrieval result is relevant or not can be made based on investigating the search results, which comprises surrogates of the videos such as image previews, preview text, user tags, comments, the video timeline, as well as reviewing the video itself. After finishing the search task, the user was instructed to review the selected videos again and to decide if the selection was appropriate. Finally, after finishing each task the user was asked, if she had achieved the search goal, how satisfied she felt with the achieved result, how helpful the search functionality was in general, and how familiar she has been with the domain of the search task. Satisfaction, helpfulness, and familiarity were measured on scale from 0 (not at all) to 4 (very much).

Table 8 shows the results of the evaluation with respect to the tests *with* exploratory search (2nd column) and the control group tests *without* exploratory search (3rd column). A number of 19 persons were participating in total, 11 of them where using the exploratory search feature, eight were involved in a control group. 72 tasks were processed with utilization of the exploratory navigation and 48 without exploratory navigation. For all 72 tasks a total number of 813 queries were issued. The control group produced 609 queries for 48 tasks. 49.3% of the tasks using the exploratory search feature were accomplished successfully by the participants. The control group accomplished only 31.8% of tasks successfully. While processing the queries, in 93.6% of queries the participants felt that it is possible to achieve the

Table 8 Results of qualitative evaluation (d = standard deviation)

	With exploratory search	Without exploratory search
# of persons	11 of 19	8 of 19
# of tasks	72	48
# of queries	813	609
Task accompl.	36 (49.3%)	14 (31.8%)
Task not accompl.	37 (50.6%)	30 (68.3%)
Motivating queries	761 (93.6%)	524 (86.0%)
Satisfaction (0–4)	1.82 (d: 1.39)	1.11 (d: 1.20)
Helpfulness (0–4)	2.29 (d: 1.42)	1.66 (d: 0.85)
Familiarity (0–4)	0.97 (d: 0.99)	1.06 (d: 0.98)
Processing time	6.2 min/task (d: 3.6 min)	7.1 min/task (d: 4.2 min)
Selected videos	168 (2.33 video/task)	96 (2.00 video/task)

search objective. In the control group for only 86.0% of the queries the participants thought that it is possible to achieve the search objective.

On a scale from 0 to 4, with exploratory search the user satisfaction was evaluated to 1.82 in the average. The control group was only satisfied with 1.11 in the average. The helpfulness of the GUI was assessed with 2.29 with exploratory search, whereas the control group achieved only 1.66. The familiarity was measured to 0.97 with exploratory search and 1.06 without. The average task processing time was observed with 6.2 minutes using exploratory search and 7.1 minutes without exploratory search. Finally, 2.33 videos per task were considered to be relevant with exploratory search, whereas 2.00 videos per task were selected without exploratory search. Table 8 shows also the standard deviations for the particular results.

Summarizing the results, the number of tasks accomplished successfully was raised to 49.3% by use of the exploratory search. The motivation of participants was significantly higher with the exploratory search feature. User satisfaction was increased by 20%, helpfulness of the GUI was increased by 15%. Processing time was improved by use with exploratory search, but not very much. Familiarity is almost constant. In general, exploratory search leads to more selected videos.

5.3 Discussion

According to our evaluation results, general GUI usability as well as the user's satisfaction with the quality of the achieved search results has been determined. The evaluation can be further refined by focusing on these two different aspects separately.

For our quantitative evaluation, the achieved levels of both recall as well as precision are relatively low. Although this is not surprising, comparing it to the results of the qualitative study raises an important question: What are the characteristics of the "exploratory" search that manage to significantly improve the support for the selected task, and why is this achieved with relatively low precision and recall? The answer can be found in the nature of the search task. All queries require the user to first find a set of entities, and for these entities find the videos in which they occur. These are also the types of tasks targeted by Freebase Parallax for browsing and exploring the freebase database. We have deliberately chosen these tasks as they are expected to be best solved by an exploratory search engine. However, this seems not to be the best choice for an objective experimental setup. Instead "standard" search tasks (e.g. find videos of barack obama) should also be considered in future work. The hypothesis would then be that the systems with exploratory search would perform better on the tasks that require "some exploration" while there is no difference for the other tasks.

A drawback of the current quantitative evaluation is also that the results are obtained independently of a search task. For different types of search tasks different types of recommendations should be provided. For example, when searching for videos about German chancellors it would simply be expected a list with the names of all the German chancellors. Understanding for which tasks which types of heuristics are needed will be clarified in future work.

Altogether, the heuristics-based recommendation of related entities to a given user query has been shown to be an integral part of the exploratory search. A detailed

overview of the presented evaluation results with regular updates can be found at: http://www.yovisto.com/evaluation.

6 Conclusion and future work

In this work, we have addressed the problem of how to deploy exploratory search for video data by using semantic search technology and demonstrated an improved exploratory search for the Yovisto video search engine along with an evaluation of the exploratory search process. We have shown how to use Linked Open Data to enable a simple exploratory search for the Yovisto video search engine. By using LOD, we were able to make implicitly existing relations among Yovisto resources explicit and to augment the ordinary keyword-based search by presenting additional related information and resources to the user via an appropriate interactive user interface.

Exploratory search is supported by heuristics based on semantic data from the LOD resources, which are used to augment direct search result with navigational information that might be also relevant for the user. We have presented heuristics based on structural and statistical features of the DBpedia RDF graph for determining related information resources and evaluated the exploratory search feature by using a comparative evaluation technique. Thus, we have been able to objectively point out the positive aspects of the exploratory search approach such as a higher task accomplishment, higher motivation and satisfaction rates.

Exploratory search is at it's early stages as a research area. Currently, there does not exist an overall accepted best-practice neither on how to realize nor on how to evaluate exploratory search. We aimed to overcome this shortcoming by strongly relating our accomplishments and methods to existing research in this area.

Although, we have obviously increased the recall of obtained results by providing an exploratory search interface, the precision of the suggested resources has to be determined by the user and her personal information needs. Another problem to deal with is caused by our multilingual approach (currently, we are working with English resources as well as with German resources simultaneously).

Improvements of the graphical user interface explicitly supporting the investigative and navigational aspect of our approach will be considered in future work. For better support in data space navigation, future work is focussed on the combination of faceted and explorative search features to satisfy the searchers curiosity and to foster serendipitous discovery.

Overall, we have implemented a first prototype for exploratory video search, which gives the user the possibility to discover resources that are usually hidden away from the user's eyes in the search engine index.

References

1. Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z (2008) DBpedia: a nucleus for a web of open data. In: Proc. of 6th int. semantic web conf., 2nd Asian semantic web conf., pp 722–735
2. Berners-Lee T (2006) Linked data. World wide web design issues http://www.w3.org/DesignIssues/LinkedData.html

3. Berners-Lee T, Hendler J, Lassila O (2001) The semantic web a new form of web content that is meaningful to computers will unleash a revolution of new possibilities. Sci Am 284(5):34–43. http://www.sciam.com/article.cfm?id=the-semantic-web&print=true
4. Bizer C, Cyganiak R, Heath T (2007) How to publish linked data on the web. http://sites.wiwiss.fu-berlin.de/suhl/bizer/pub/LinkedDataTutorial/
5. Bizer C, Heath T, Idehen K, Berners-Lee T (2008) Linked data on the web. In: Proc. of the 17th int. conf. on world wide web, ACM, pp 1265–1266
6. Bollen J, Nelson ML, Geisler G, Araujo R (2007) Usage derived recommendations for a video digital library. J Netw Comput Appl 30(3):1059–1083
7. Borlund P (2003) The concept of relevance in IR. J Am Soc Inf Sci Technol 54(10):913–925
8. Borlund P (2003) The IIR evaluation model: a framework for evaluation of interactive information retrieval systems. Inf Res 8(3). http://informationr.net/ir/
9. Brickley D, Miller L (2007) FOAF vocabulary specification 0.91. Online at: http://xmlns.com/foaf/spec/
10. Chen X, Zhang C (2006) An interactive semantic video mining and retrieval platform - Application in transportation surveillance video for incident detection. In: ICDM 2006: proc. of 6th IEEE int. conf. on data mining, IEEE Computer Soc., Hong Kong, pp 129–138
11. Christel MG (2008) Supporting video library exploratory search: when storyboards are not enough. In: Proc. of the int. conf. on content-based image and video retrieval, ACM, New York, NY, USA, pp 447–456
12. Dan Brickley R (2004) RDF Vocabulary Description Language 1.0: RDF Schema. Tech. rep., W3C
13. Day N, Martínez JM (2000) Introduction to MPEG-7. Tech. Rep. ISO/IECT JTC1/SC29/WG11 N3751, International Organisation for Standardisation
14. Duke A, Heizmann J (2009) Semantically enhanced search and browse. Semantic Knowledge Management pp 85–102
15. Fouss F, Saerens M (2008) Evaluating performance of recommender systems: an experimental comparison. In: Proceedings of the 2008 IEEE/ACM international joint conference on web intelligence (WIC 2008), pp 735–738
16. Garcia R, Celma O (2005) Semantic integration and retrieval of multimedia metadata. In: Proc. of 4rd int. semantic web conf. knowledge markup and semantic annotation workshop, Galway, Ireland
17. Gruber TR (1995) Toward principles for the design of ontologies used for knowledge sharing. Int J Hum-Comput Stud 43(5–6):907–928
18. Guha R, McCool R, Miller E (2003) Semantic search. In: WWW '03: proc. of the 12th int. conf. on world wide web, ACM Press, New York, NY, USA, pp 700–709
19. Hu J, Wang G, Lochovsky F, Sun JT, Chen Z (2009) Understanding user's query intent with wikipedia. In: WWW '09: proc. of the 18th international conference on world wide web, ACM, New York, NY, USA, pp 471–480
20. International Organization for Standardization: Information and Documentation – The Dublin Core Metadata Element Set. (2003) ISO 15836
21. Ivan Herman Ralph Swick DB (2004) Resource description framework (RDF). Tech. rep., W3C
22. Lo EHS, Pickering MR, Frater MR, Arnold JF (2009) Query by example using invariant features from the double dyadic dual-tree complex wavelet transform. In: Proc. of the ACM int. conf. on image and video retrieval (CIVR '09), ACM, New York, NY, USA, pp 1–8
23. Mangold C (2007) A survey and classification of semantic search approaches. In: Int. J. Metadata, Semantics and Ontology, vol 2, pp 23–34
24. Marchionini G (2006) Exploratory search: from finding to understanding. Commun ACM 49(4):41–46
25. Meij EJ, Bron M, Huurnink B, Hollink L, de Rijke M (2009) Learning semantic query suggestions. In: 8th Int. semantic web conf. (ISWC 2009), Springer
26. Miles A, Bechhofer S (2008) Skos simple knowledge organization system reference. World Wide Web Consortium, Working Draft
27. Milne D, Witten IH (2008) Learning to link with wikipedia. In: CIKM '08: proceeding of the 17th ACM conference on information and knowledge management, ACM, New York, NY, USA, pp 509–518
28. Oren E, Delbru R, Catasta M, Cyganiak R, Stenzhorn H, Tummarello G (2008) Sindice.com: a document-oriented lookup index for open linked data. IJMSO 3(1):37–52

29. Petratos P (2008) Informing through user-centered exploratory search and human-computer interaction strategies. Issues in Informing Science and Information Technology (IISIT) 5:705–727. Information Science Institute, 131 Brookhill Court, Santa Rosa, California 95409 USA
30. Pollitt AS, Ellis GP, Smith MP (1994) HIBROWSE for bibliographic databases. J Inf Sci 20(6):413–426. Chartered Institute of Library and Information Professionals, doi:10.1177/016555159402000604
31. Prud'hommeaux E, Seaborne A (2008) SPARQL query language for RDF. W3C
32. Qu Y, Furnas GW (2008) Model-driven formative evaluation of exploratory search: a study under a sensemaking framework. Inf Process Manag 44(2):534–555
33. Richard Newman Danny Ayers SR (2005) Tag Ontology http://www.holygoat.co.uk/owl/redwood/0.1/tags/
34. Sack H, Waitelonis J (2006) Integrating social tagging and document annotation for content-based search in multimedia data. In: Proc. of the 1st semantic authoring and annotation workshop, Athens (GA), USA
35. Sack H, Waitelonis J (2006) Automated annotation of synchronized multimedia presentations. In: Proceedings of the ESWC 2006 workshop on mastering the gap: from information extraction to semantic representation, CEUR Workshop Proceedings
36. Schraefel M, Wilson M, Russell A, Smith DA (2006) mSpace: improving information access to multimedia domains with multimodal exploratory search. Commun. ACM 49(4):47–49
37. Schreiber G, Amin A, Aroyo L, van Assem M, de Boer V, Hardman L, Hildebrand M, Omelayenko B, van Osenbruggen J, Tordai A, Wielemaker J, Wielinga B (2008) Semantic annotation and search of cultural-heritage collections: the MultimediaN E-Culture demonstrator. Semantic Web Challenge 2006/2007; Web Semantics: Science, Services and Agents on the World Wide Web 6(4):243–249. doi:10.1016/j.websen.2008.08.001
38. Semantic Web Education and Outreach Interest Group (2009) http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData/
39. Singh H, Cheung A, Guadarrama S, Loer C, Nikravesh M (2006) Evaluating ontology based search strategies. In: Soft computing for information processing and analysis, vol 164, pp 189–202
40. Smeaton AF, Over P, Kraaij W (2006) Evaluation campaigns and TRECVid. In: MIR '06: proceedings of the 8th ACM international workshop on multimedia information retrieval, ACM, New York, NY, USA, pp 321–330
41. Smeulders AWM, van Gemert JC, Huurnink B, Koelma DC, de Rooij O, van De Sande KEA, Snoek CGM, Veenman CJ, Worring M (2007) Semantic video search. In: Proc. of 14th int. conf. on image analysis and processing, IEEE Computer Soc, pp 51–58
42. Smith MK, Welty C, McGuinness DL (2004) OWL Web Ontology Language Guide. Tech. rep., W3C, World Wide Web Consortium. http://www.w3.org/TR/owl-guide/
43. Stefaner M, Urban T, Seefelder M (2008) Elastic lists for facet browsing and resource analysis in the enterprise, dexa. In: 19th international conference on database and expert systems application, pp 397–401
44. Tran DT, Bloehdorn S, Cimiano P, Haase P (2007) Expressive resource descriptions for ontology-based information retrieval. In: Proc. of the 1st int. conf. on the theory of information retrieval (ICTIR'07)
45. Waitelonis J, Sack H (2009) Augmenting video search with Linked Open Data. In: Proc. of int. conf. on semantic systems 2009
46. Waitelonis J, Sack H (2009) Towards exploratory video search using linked data. In: ISM '09: proc. of the 2009 11th IEEE int. symp. on multimedia, IEEE Computer Society, Washington, DC, USA, pp 540–545
47. Yee KP, Swearingen K, Li K, Hearst M (2003) Faceted metadata for image search and browsing. In: CHI '03: proceedings of the SIGCHI conference on human factors in computing systems, ACM, New York, NY, USA, pp 401–408

Jörg Waitelonis is Research Assistant at the Hasso Plattner-Institute for IT-Systems Engineering (HPI) at the University of Potsdam. After graduating in computer science at the Friedrich-Schiller-University in Jena in 2006 he developed the video search engine yovisto.com. Yovisto started as an ESF/BMWi (European Social Fund/German Government) funded project with the objective to develop a video search engine for academic lecture recordings and was relocated from Friedrich-Schiller-University Jena to Hasso Plattner-Institute to become a research platform for semantic multimedia technologies. Furthermore, Jörg worked on the multimedia processing system REPLAY at Swiss Federal Institute of Technology Zürich.

Harald Sack is Senior Researcher at the Hasso Plattner-Institute for IT-Systems Engineering (HPI) at the University of Potsdam. After graduating in computer science at the University of the Federal Forces Munich Campus in 1990, he worked as systems/network engineer and project manager in the signal intelligence corps of the German federal forces from 1990–1997. In 1997 he became an associated member of the graduate program 'mathematical optimization' at the University of Trier and graduated with a PhD thesis on formal verification in 2002. From 2002–2008 he did research and teaching as a postdoc at the Friedrich-Schiller-University in Jena and since 2007 he has a visiting position at the HPI. His areas of research include multimedia retrieval, semantic web, knowledge representations and semantic enabled retrieval. Since 2008 he is speaker of the special interest group 'multimedia- and hypermediasystems' of the german computer science society (Gesellschaft für Informatik) and general secretary of the german IPv6 council.